APPENDIX A
CLAIMS

1. (Amended) A method for achieving client to server end to end security guarantees, comprising:

employing a proxy between a client and a server to provide connection links between said client and said server;

embedding a secure coprocessor for use as an agent of said client and/or said server which assures that said proxy cannot tamper with the functioning of said agent, said agent being a software program or hardware logic operating within the confines of said coprocessor;

said coprocessor is located at the site of said proxy and: (a) acts as a converter between at least one protocol said client supports, and at least one other protocol supported by said server, ~~(b) assures that said proxy cannot tamper with the functioning of said agent,~~ ~~(c)~~ (b) guarantees that an application embedded in said coprocessor performs to a degree of security proscribed by said client and/or said server; and

employing the respective security protocols of said at least one protocol and said at least one other

Claims 2 - 4 (Canceled)

Claim 5 (Previously presented) A method as recited in claim 1 wherein the client is a pervasive computing device.

Claim 6 (Previously presented) A method as recited in claim 1, further comprising the step of adapting content supplied by the client to fit constraints of the server and/or the connection links.

Claim 7 (Previously presented) A method for providing secure communications on a network, the method comprising:

securely embedding an agent at a proxy in the network, and,

splicing a plurality of secure communication protocols of different protocol suites into the agent.

Claim 8 (Currently Amended) A method as recited in claim 7 wherein the step of splicing a security protocol of ~~the~~ a Wireless Application Protocol Suite (WAP) to that of ~~the~~ a Internet Protocol (IP) device.

Claim 9 (Previously presented) A method as recited in claim 8 wherein the Wireless Application Protocol suite is used by a pervasive computing device.

B1

Claim 10 (Previously presented) A method as recited in claim 7, further comprising the agent performing at least one content adaptation function.

Claim 11 (Previously presented) A method as recited in claim 10, wherein the step of performing includes maintaining communication privacy.

Claim 12 (Previously presented) A method as recited in claim 7, further comprising maintaining a state of splicing process resulting from the step of splicing.

Claim 13 (Previously presented) A method as recited in claim 12, wherein the step of maintaining includes employing a storage device external to the proxy, and using cryptographic means to encrypt the state.

Claim 14 (Previously presented) A method for providing network security to a network employing a proxy, the method comprising:

embedding a trusted application in a secure coprocessor located at the site of a proxy; and

delegating to a network infrastructure a task of enforcing a trust model.

Claim 15 (Previously presented) A method as recited in claim 14, further comprising guaranteeing that the application is trusted to enforce th trust model between at least one server and a plurality of clients.

Claim 16 (Previously presented) A method as recited in claim 14, further comprising assuring the tamper resistance of the application.

B1

Claim 17 (Previously presented) A method for secure communication between a client and a server employing an untrusted proxy; the method comprising:

embedding a coprocessor at the proxy;

the proxy receiving a specific communication request from a client;

the proxy forming an n-tuple for the specific communication;

the proxy forwarding the n-tuple to the coprocessor;

the coprocessor generating a response, including a directive, to the n-tuple;

the coprocessor sending the response to the proxy, and

the proxy implementing the directive.

Claim 18 (Previously presented) A method of claim 17, wherein the coprocessor is a secure coprocessor.

Claim 19 (Previously presented) A method of claim 17, wherein the step of receiving includes:

awaiting a connection request from a client;

creating an entry in a storage module for the client;

determining a sender of each received packet; and

retrieving a stored entry.

Claim 20 (Previously presented) A method of claim 19, wherein the n-tuple includes a sender id, an entry from a storage module and the received packet.

Claim 21 (Previously presented) A method of claim 17, wherein the client and the server can be either a sender or a receiver, and the step of generating includes employing a first protocol from the sender to the proxy and a second protocol from the proxy to the receiver and translating between the first and second protocols.

Claim 22 (Previously presented) A method of claim 21, wherein the translating includes decrypting the received packet as specified by the security parameters negotiated as per the first protocol and encrypting the decrypted packet as specified by the security parameters of the second protocol.

Claim 23 (Previously presented) A method of claim 21, wherein the translating includes modifying the received packet to meet constraints of the receiver and wherein the directive includes forwarding to the receiver the packet resulting from the step of modifying.

Claim 24 (Previously presented) A method as recited in claim 23, further comprising aggregating a plurality of packets into a group of packets and performing content adaptation on the group of packets.

Claim 25 (Previously presented) A method of claim 17, wherein the communication between the client and the proxy employ protocols specified by the Wireless Application Protocol suite (WAP).

Claim 26. (Currently Amended) A system to control security of a proxy interconnecting a client to a server, comprising:

a secure coprocessor, used as an agent of the client and/or a server, said secure coprocessor being located at the site of said proxy ; said agent being a software program or hardware logic operating within the confines of said coprocessor and

B1

an application embedded in said secure coprocessor which acts as a converter between at least one protocol said client supports and at least one other protocol supported by said server, wherein said secure coprocessor employs respective security protocols of said at least one protocol and said at least one other protocol; said secure coprocessor also assuring that said proxy cannot tamper with the functioning of said agent, and guaranteeing that an application embedded in said coprocessor performs to a degree of security proscribed by said client and/or said server.

Claims 27 - 29 (Canceled)

Claim 30 (Previously presented) A system as recited in claim 26, wherein the application embedded in the coprocessor adapts content supplied by the server to fit constraints of the client and the connection links.

Claim 31 (Previously presented) A system as recited in claim 26, wherein the application embedded in the coprocessor adapts content supplied by the client to fit constraints of the server and the connection links.

Claim 32 (Previously presented) A system for providing network security to a network employing a proxy, the system comprising:

a secure coprocessor located at the site of a proxy; and

a trusted application embedded in the coprocessor wherein the coprocessor delegates the task of enforcing an arbitrary trust model to the application.

Claim 33 (Previously presented) A system as recited in claim 32, wherein the coprocessor functions to guarantee that the application is trusted to enforce the trust model between at least one server and a plurality of clients.

Claim 34 (Previously presented) A system as recited in claim 32, wherein the coprocessor functions to assure the tamper resistance of the application.

Claim 35. (Currently Amended) An article of manufacture comprising a computer usable medium having computer readable program code means embodied therein for achieving client to server end to end security guarantees, the computer readable program code means in said article of manufacture comprising computer readable program code means for causing a computer to effect:

employing a proxy between a client and a server to provide connection links between said client and said server;

embedding a secure coprocessor for use as an agent of said client and/or said server which assures that said proxy cannot tamper with the functioning of said agent, said agent being a software program or hardware logic operating within the confines of said coprocessor;

said coprocessor is located at said proxy and: (a) acts as a converter between at least one protocol said client supports, and at least one other protocol supported by said server, ~~(b) assures that said proxy cannot tamper with the functioning of said agent,~~ ~~(c)~~ (b) and guarantees that an application embedded in said coprocessor performs to a degree of security proscribed by said client and/or said server;

B1

employing the respective security protocols of said at least one protocol and said at least one other protocol.

Claim 36 (Previously presented) An article of manufacture as recited in claim 35, the computer readable code means in said article of manufacture further comprising computer readable program code means for causing a computer to effect the coprocessor assuring that the proxy can not tamper with the functioning of the agent.

Claim 37 (Canceled)

Claim 38. (Currently Amended) An article of manufacture comprising a computer usable medium having computer readable program code means embodied therein for achieving client to server end to end security guarantees, the computer readable program code means in said article of manufacture further comprising computer readable program code means for causing a computer to effect:

employing a proxy between a client and a server to provide connection links between said client and said server;

embedding a secure coprocessor for use as an agent of said client and/or said server which assures that said proxy cannot tamper with the functioning of said agent, said agent being a software program or hardware logic operating within the confines of said coprocessor;

said coprocessor is located at said proxy site and: (a) acts as a converter between at least one protocol said client supports, and at least one other protocol supported by said server, ~~(b) assures that said proxy cannot tamper with the functioning of said agent,~~ ~~(c)~~ (b) adapts content supplied by said server to fit constraints of said client and/or connection links.

employing the respective security protocols of said at least one protocol and said at least one other protocol.

Claim 39. (Previously presented) An article of manufacture comprising a computer usable medium having computer readable program code means embodied therein for achieving client to server end to end security guarantees, the computer readable program code means in said article of manufacture further comprising computer readable program code means for causing a computer to effect:

employing a proxy between a client and a server to provide connection links between said client and said server;

embedding a secure coprocessor for use as an agent of said client and/or said server;

said coprocessor is located at said proxy site and: (a) acts as a converter between at least one protocol said client supports, and at least one other protocol supported by said server, b) assures that said proxy cannot tamper with the functioning of said agent, and (c) adapts content supplied by said server to fit constraints of said server and connection links;

employing the respective security protocols of said at least one protocol and said at least one other protocol.

Claim 40. (Previously Presented) A computer program product comprising a computer usable medium having computer readable program code means embodied therein for providing secure communication on a network, the computer readable program code means in said computer program product comprising computer readable program code means for causing a computer to effect:

securely embedding an agent at the site of a proxy in the network, and

splicing a security protocol of a Wireless Applications Protocol suite (WAP) to that of the Internet Protocol (IP) suite.

Claim 41 (Canceled)

Claim 42. (Previously presented) A computer program product comprising a computer usable medium having computer readable program code means embodied therein for providing secure communication on a network, the computer readable program code means in said computer program product comprising computer readable program code means for causing a computer to effect:

securely embedding an agent at a proxy in the network, and

splicing a plurality of secure communication protocols of different protocol suites into said agent, wherein said splicing includes maintaining end to end security guarantees at said server.

B1

43. (Amended) A computer program product comprising a computer usable medium having computer readable program code means embodied therein for providing secure communication on a network, the computer readable program code means in said computer program product further comprising computer readable program code means for causing a computer to effect:

securely embedding an agent at a proxy in the network, and

said agent performing at least one content adaptation function;

splicing a plurality of secure communication protocols of different protocol suites into said agent.

Claim 44. (Previously presented) A computer program product comprising a computer usable medium having computer readable program code means embodied therein for providing secure communication on a network, the computer readable program code means in said computer program product further comprising computer readable program code means for causing a computer to effect :

securely embedding an agent at a proxy in the network, and

splicing a plurality of secure communication protocols of different protocol suites into said agent;

B1

maintaining a state of said splicing process resulting from said step of splicing, wherein said step of maintaining includes employing a storage device external to said proxy, and using cryptographic means to encrypt the state of a splicing process resulting from the step of splicing.

Claim 45 (Canceled)

Claim 46 (Previously presented) A computer program product comprising a computer usable medium having computer readable program code means embodied therein for providing network security to a network employing a proxy, the computer readable program code means in said computer program product comprising computer readable program code means for causing a computer to effect the steps of :

embedding a trusted application in a secure coprocessor located at the site of a proxy; and

delegating to a network infrastructure a task of enforcing a trust model.

Claim 47 (Previously presented) A computer program product as recited in claim 46, the computer readable program code means in said computer program product further comprising computer readable program code means for causing a computer to effect the step of guaranteeing that the application is trusted to enforce the trust model between at least one server and a plurality of clients.

Claim 48 (Previously presented) A computer program product as recited in claim 46, the computer readable program code means in said computer program product further comprising computer readable program code means for causing a computer to effect the step of assuring the tamper resistance of the application.

Claim 49 (Previously presented) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for secure communication between a client and a server employing an untrusted proxy, said method steps comprising:

embedding a coprocessor at the proxy;

the proxy receiving a specific communication request from a client;

the proxy forming an n-tuple for the specific communication;

the proxy forwarding the n-tuple to the coprocessor;

the coprocessor generating a response, including a directive, to the n-tuple;

the coprocessor sending the response to the proxy, and

the proxy implementing the directive.

Claim 50 (Previously presented) A program storage device readable by machine as recited in claim 49, wherein the coprocessor is a secure coprocessor.

Claim 51 (Previously presented) A program storage device readable by machine as recited in claim 49, wherein the step of receiving includes:

awaiting a connection request from a first client;

creating an entry in a storage module for the client;

determining a sender of each received packet;

retrieving a stored entry.

Claim 52 (Previously Presented) A program storage device readable by machine as recited in claim 49, wherein the n-tuple includes a sender id, an entry from a storage module and the received packet.

Claim 53 (Previously Presented) A program storage device readable by machine as recited in claim 49, wherein the client and the server can be either a sender or a receiver, and the step of generating includes employing a first protocol from the sender to the proxy and a second protocol from the proxy to the receiver and translating between the first and second protocols.

Claim 54 (Previously Presented) A program storage device readable by machine as recited in claim 49, wherein the translating includes decrypting the received packet as specified by the security parameters negotiated as per the first protocol and encrypting the decrypted packet as specified by the security parameters of the second protocol.

Claim 55 (Previously Presented) A program storage device readable by machine as recited in claim 49, wherein the translating includes modifying the received packet to meet constraints of the receiver and wherein the directive includes forwarding to the receiver the packet resulting from the step of modifying.

Claim 56 (Previously Presented) A program storage device readable by machine as recited in claim 55, said method steps further comprising the step of aggregating a plurality of packets into a group of packets and performing content adaptation on the group of packets.

Claim 57 (Previously Presented) A program storage device readable by machine as recited in claim 49, wherein the communication between the client and the proxy employ protocols specified by the Wireless Application Protocol suite (WAP).

Claim 58 (Previously presented) A method as recited in claim 1, further comprising the step of the coprocessor adapting content supplied by the server to fit constraints of the client and/or the connection links.

Claim 59. (Previously presented) A method as recited in claim 7, wherein the splicing includes maintaining end to end security guarantees without a modification to a server involved in the communication.

translation. These may be achieved with varying degrees of the following guarantees:

(a) Given that a certain functionality is intended and expected of the application executing on the secure coprocessor, an external ~~agent~~ entity can neither subvert nor disrupt the execution of such an application.

(b) In typical network security protocols consisting of a client and a server, the trust model guarantees that an untrusted third party can neither eavesdrop nor subvert the communication between the client and the server.

In one embodiment, the same guarantees of such a trust model are obtained even though communication occurs indirectly through the application executing in the secure coprocessor at the site of the proxy.

(c) In some alternate embodiments, the same trust guarantees are obtained by varying levels of modifying the server to support the same protocols as the client. An advantage provided by the method of the present invention is that it is advantageous that no modification be required at the client or the server even if they do not support the same set of protocols.

Another aspect of the present invention is to provide a means by which a server can securely delegate its